Re-Engineering Paper Technical Manual's Troubleshooting Procedures

Dr. Li Pi Su, Advanced Technology Office
The US Army Test, Measurement, Diagnostic Equipment Activity
The US Army Aviation and Missile Command, Redstone Arsenal, AL 35898
Ms. Mary Nolan, Giordano Automation Corp. Sparta, NJ 07871

Abstract -- As the Department of Defense (DOD) downsizes there is a great need to reduce the cost and manpower burden associated with maintenance of weapon systems. Traditionally, technical manuals used for field maintenance of DOD systems have relied heavily on troubleshooting procedures, which are presented in "flow chart" format of fault trees. These flow charts guide the maintainer through test procedures to isolate parts that cause equipment malfunction. These procedures are static, that is, they are highly structured around a predetermined sequence of tests, do not become "smarter" over time with historical maintenance data, and only take into account those symptoms and faults which the original developer considered. They are often incomplete, sometimes wrong, and are very difficult to update and maintain. As maintenance evolved into the computer-assisted age, a major opportunity exists to significantly enhance the technical manuals, the basic logic, and information/knowledge representation underlying troubleshooting procedures.. This paper provides the high lights of research and development results on the technical aspects as how to efficiently transition from flow-chart intensive knowledge representation to a knowledge-based system. The results of reengineering the legacy trouble-shooting procedures provides, at least, the following benefits: (1) replacing fault trees with knowledge based reasoning about faults related to symptoms; (2) providing the capability to dynamically relate faults to symptoms; (3) equipping the ability to use historical maintenance data to continuously improve maintenance capability; (4) providing more user-friendly interactive electronic technical manuals; and (5) providing the ability to house "expert" diagnostics information in a form that becomes usable and available to novice technicians.

I. INTRODUCTION

Currently, field maintenance of Department of Defense (DOD) weapon systems relies heavily on troubleshooting procedures (TPs). However, these TPs are in a "flow chart" format of fault trees. The TPs are static, that is, they are highly structured around a predetermined sequence of tests and they do not grow smarter over time with the use of maintenance historical data. Moreover, the TPs are sometimes incomplete and/or inaccurate in performing diagnostics. The TPs are also cumbersome and poorly integrated with embedded test and automated test procedures. These factors result in a significant maintenance burden for the DOD.

To cope with the DOD's shrinking budget and downsizing, reengineering TPs to reduce this maintenance burden is a must.

The Advanced Technology Office (ATO), U.S. Army Test, Measurement, Diagnostic Equipment Activity (USATA), U.S. Army Aviation and Missile Command, initiated a research and development (R&D) effort to reengineer the TPs in 1995. In December 1996, the ATO

teamed up with the Logistics Support Engineering Directorate, ARDEC and Giordano Automation Corporation (GAC) to investigate and develop an enhancement methodology for reengineering the technical manual's TPs. This paper provides some of the key results of our efforts.

II. ISSUES WITH TPs AND TECHNICAL MANUALS (TMs)

In general, the TPs contained within TMs have following issues that make them noneffective and result in a big maintenance burden:

A. Most troubleshooting logic is represented in flow-chart format of "Fault Tree". The troubleshooting logic flows as follows: The results of one test leads to execution of the next test, whose result in turn, leads again to the next test. In short, the sequence of tests are predetermined and inflexible.

B. Fault trees are developed as part of the TM development process and are based upon a specified set of fault conditions. They are static, that is, they do not adapt to new or unplanned fault modes and they do not improve over time, based upon field experience.

C. Fault trees' static logic is limited to a "single fault assumption" and in multiple fault situations, can become very unreliable. This results in incomplete and/or inaccurate in diagnosing.

D. The TPs and TMs are often very cumbersome and poorly integrated with embedded test and automated test procedures. This results in ineffective maintenance or no maintenance solution.

III. ENHANCEMENT GOAL

The goal of this R&D efforts is to significantly improve the TPs and TMs to reduce mean-time-to repair, maintenance training requirements, and maintenance costs. For field diagnostics, the goal is to reengineer static TPs into a much more robust, simpler, and dynamic fault isolation mechanism. For the TMs, integration with embedded test and automated test procedures must be enhanced be more interactive and user-friendly.

IV. TECHNICAL APPROACHES

To accomplish the goal, a new paradigm of diagnostics reasoning technology is required that has the capability to eliminate complex diagnostic logic paths and the capability of receiving and dynamically interpreting any test results from any source, in any order, and with as many or as few test results at a time [2,3,4,5]. Moreover, an Interactive Electronic Technical Manual (IETM) authoring and display system must be robust and be able to integrate the diagnostics logic, the test procedures and routines, and technical manual information to provide a truly interactive interface between users, systems, test resources, and maintenance functions [6].

For these purposes, the technical approach requires, firstly, a diagnostics mechanism wherein **diagnostics logic is an entity independent of test procedures and routines, that is able to dynamically interpret test results,** and whose **diagnostics logic can be recaptured from legacy TPs and other data from TMs**.

Secondly, it requires an **overall object-oriented structure and an open architecture.** The concept of object-oriented programs enable taking full advantage of dealing with the diagnostic logic as an entity independent of test procedures and routines. The independent diagnostic logic contained in the software object can be rehosted to any platform without any problem, because it is simply a software object - such as a binary file [4,5].

Thirdly, it also requires an IETM authoring and display package, which is CALS-compliant and also highly robust. This authoring system must be able to integrate the diagnostics logic, the test procedures and routines, technical manual information, and other required maintenance information to provide truly interactive and full complement of dialogs.

In looking for a new paradigm of diagnostics technology satisfying above requirements, **Diagnostician** was identified. The **Diagnostician** is a greatly enhanced model-based diagnostic reasoning system that is significantly enhanced from DARTS, a concept/technology developed by the ATO, USATA, and GAC during 1992-1993 [1] and commercialized by the GAC. The **Diagnostician**'s model-based diagnostic software is used in an object-oriented software environment - the diagnostics logic is an independent entity that is able to dynamically interpret test results. The basic knowledge used by the **Diagnostician** includes a fault propagation model that maps all possible faults to all possible symptoms (tests or visual observations). Normally, the basis for this mapping is derived (automatically translated) from computer aided design (CAD) data (netlists describing, hierarchy, interconnectivity, components and pins, and signal flow).

The **Diagnostician**'s knowledge bases, called a fault/symptom matrix, is a model in the form of a connectivity matrix that represents the propagation of faults (rows in the matrix) to observable

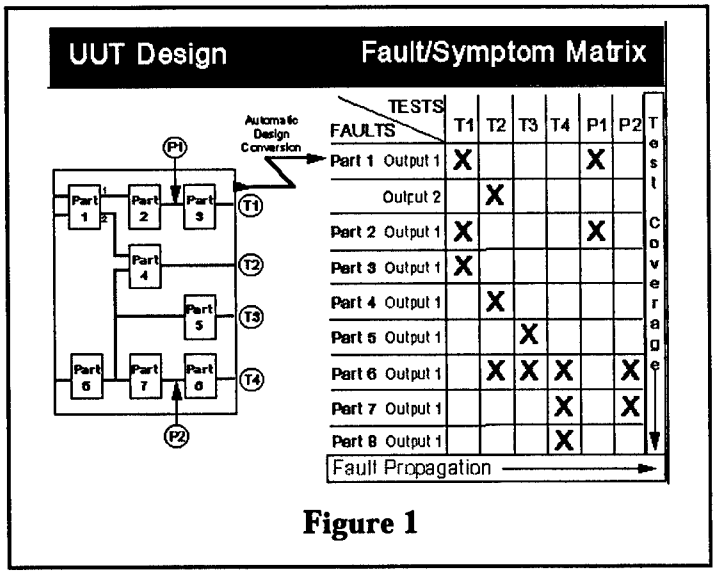

Figure 1

measurement locations and the coverage of tests that Pass or Fail (columns in the matrix). (See Figure 1). When used in run-time, the **Diagnostician**'s algorithms and knowledge bases (matrix) operate to isolate fault without troubleshooting sequences and without hard-code diagnostic logic.

To identify the required IETM authoring system, a study was conducted by the ATO over four authoring systems. A chapter of a paper technical manual was selected and converted into an IETM using each authoring tool. This was done to evaluate the features of each tool [6]. Sixteen different features of these authoring tools were investigated. These features include spell checking, animation, audio, video, execution of external programs, implementation of hotkeys for voice actuation, ease of linking frames, hotlinks capabilities, flexibility, whether or not a standardized format is required, whether the authored text requires compilation before viewing, graphics requirements, Object Linking and Embedding, cost, provisions for DCA, J1708, J4 and 1553 connections and interfacing and the troubleshooting capabilities of the IETMs.

The Raytheon (formerly Hughes) "Advanced Integrated Maintenance Support System (AIMSS)" IETM authoring and display system was identified to be the best IETM authoring system for this development effort. The AIMSS is a CALS-compliant and Class IV SGML IETM system. The AIMSS uses overall object-oriented structure and is an open architecture approach. It can integrate the overall IETM with diagnostics logic, test procedures and routines, technical manual information, and other requirements. AIMSS uses a common graphical user interface to display maintenance information in text, graphics, and table windows that can be easily manipulated by the technician for preferred viewing. Hypertext and graphic "hot spots" are embedded in descriptions and procedures to provide rapid access to related information contained in the database.

V. REENGINEERING PROCESS

The objective of the R&D efforts is to identify/develop a methodology/process to significantly improve the TPs and TMs to reduce mean-time-to repair, maintenance training requirements, and to reduce maintenance costs. As mentioned above, the **Diagnostician** will enhance legacy TPs by using dynamics model-based reasoning. Normally, the **Diagnostician**'s diagnostics knowledge bases are derived from the systems' design information. For legacy systems, the paper TMs and their TPs are the primary source of information that pertains to diagnostics. The issue now is how to cost-effectively reengineer the TPs into dynamic knowledge bases for use with the **Diagnostician**. The solution is to implement the following three steps cost effectively: (1) **Capture Diagnostics Logic from Paper TMs**; (2) **Reengineer Test Ordering Constraints and Information**; and (3) **Seemlessly Integrate Experiences into Model Based Diagnoses.**

A. Capture Diagnostics Logic from Paper TMs

Normally, and optimally, the diagnostic knowledge base is automatically generated from CAD output files, or netlists. A primary benefit of **Diagnostician** is its model-based diagnostics approach--a design-derived knowledge base that is a one-for-one *representation* of the design. That is, the direct relationship between design and fault propagation. For legacy TMs, this does not exist and even the netlist data may not exist. Therefore, alternate means for data import are required to reconstruct the relationships between symptoms and faults. In reengineering TMs' TPs, the best we can hope to accomplish, initially, is to generate a diagnostic representation that is as good as the existing TPs, because data is limited and we must assume that further information is unavailable. Note that a fault tree is an *interpretation,* and the reengineering of that fault tree into a knowledge base is a further interpretation of that interpretation.

There are four alternatives to capture legacy diagnostic data. The users can use any one or more to cost-effectively capture the diagnostic data.

1. **Capture SGML Tagged Data** - Automated software routines were written to extract diagnostic logic from SGML tagged data. The Army's 2361 DTD was analyzed with respect to data content of troubleshooting logic. A standard format was developed that allows conversion tools to be tailorable to other DTDs, since the basic content is expected to be the same.

2. **Capture From Paper-Based Troubleshooting Trees** - A reengineering methodology and supporting tools have been developed to analyze the fault tree structure representation of TPs and represent that tree structure in a knowledge base format. The reengineering methodology includes generating the diagnostic model, defining each test procedure, and incorporating the test path to each fault using the tools inside the Diagnostic Profiler.

3. **Capture from an "Expert"** - If an engineer or technician having expertise on the system

application is available, that expert can create the diagnostic model by directly authoring the fault conditions and correlating the test results or symptoms to those fault conditions (define how tests "cover" faults).

4. **Capture from Schematic Data** - If detailed schematic data is available, as well as knowledge on test coverage, the user can use a schematic capture program, such as OrCAD, to enter schematics and output EDIF netlist files. These files can be directly imported to the Diagnostic Profiler.

B. Reengineer Test Ordering Constraints and Information

In most TPs, test sequences are predetermined by fault tree structures. The tree structures represent the results of an analysis rather than the underlying information from which they were generated.

During the R&D efforts, it was determined that additional test ordering services must be added to the Diagnostician to make it truly serviceable for reengineering legacy TMs' TPs. Two factors that provide overriding test order constraints are (1) the impossibility of implementing tests under certain circumstances (e.g., measuring a frequency on a signal with no amplitude or checking a powerless display for fault codes), and (2) the need to use tests to step a unit under test through each of its states (e.g., power up, set mode, start function, handle error, etc.) especially when stepping from one state to another takes a relatively long time.

Based on these findings, a set of conditions and associated constraints or actions was developed to be used in generating tests dynamically. Each test could have no conditions placed on it, conditions about the status of one or more tests placed on it, or conditions about the status of all the other tests placed on it. Test conditions and constraints include whether the test is *permitted* to be run, whether it is *prohibited* from being run and whether it is *forced* to be run based upon the outcome of previously run tests.

C. Seemlessly Integrate Experience Data into the Model Based Diagnostics

Now, the legacy TPs have been reengineered into model based diagnostic logic, inferences are structured in an object-oriented, database-type software architecture, and the underlying data can be easily improved and updated over time. With these enhancements, run-time diagnostics can be "smartened" by incorporating field information into appropriated diagnostic reasoning. There are many useful data sources and situations that can smarten diagnoses: Frequency of Parts' Failures; Failure Modes; New or Unforeseen/Un-Modeled Failure Modes; New or Unforeseen/ Un-Profiled Test Data; and Mismatches in Diagnostic Knowledge Base (DKB) based on Incorrect Design Data Modeling and/or Test Coverage Input, etc.

To make the best use of the historical/field data mentioned above, the overall implementation strategy is to collect data on a local basis and allow this local data to be automatically

incorporated into the diagnostic reasoning process for current/local diagnostic sessions. The experience data, all local history, would be collected and processed/coordinated by a central facility. The historical data would be analyzed with automated tools and the system DKB would be updated accordingly. Then, processed/coordinated data will be redistributed to each using site.

The refinement and maturation of the DKB can be performed as follows:

1. **Frequency of Parts' Failures and Failure Modes (Failure Probability Updates)** - The DKB contains relative failure rate information that is used to determine the most probable cause of a set of symptoms. For diagnostic sessions where there is an ambiguity group, failure probability data is used to weigh the most probable suspect.

A "Run-time Smartener (RTS)" has been developed to incorporate failure rate data from actual field experience. The RTS utilizes the original failure rate data as initial Bayesian probabilities and updates these probabilities from a log file each time that a repair has been identified to correct a problem. User input has been minimized.

To implement the RTS, a Fault History Database is extracted from the historical maintenance records/database, or directly from Diagnostician data logging mechanisms. At the local level, mechanisms have been incorporated to the Diagnostician to automatically update fault probability data based on the field experience indicated by a Fault History Database. This is accomplished through a secondary file containing updated probability data. Incorporation of these updates into the primary DKB is performed at the centralized maintenance activity. The updated DKB is then distributed to local users.

2. **New or Unforeseen/Un-modeled Failure Modes** - To improve diagnoseability for the new or unforeseen/un-modeled failure modes, an Observable History Database (OHD) is created. An OHD is a database that contains any faults that were unaccounted for by the DKB in a maintenance session. It is extracted from the historical maintenance records/database as a secondary file accessed by the Diagnostician during a diagnostic session. The OHD would reflect that fault mode. The symptoms that were observed or tests that were measured will be noted and the knowledge base structure filled in accordingly. The fault data would be denoted as a learned mode. Certain symptom data may need to be noted as "masked" for uncertain test data. For these learned modes, in some cases, it may not be appropriate to use pass data to eliminate a fault mode in the reasoning process, since the confidence in the test coverage data is more uncertain (not physically tied to the model.) If this is so, only fail test data is used for reasoning, not pass. Some sort of a confidence level should be attributable to the learned fault mode.

3. **New or Unforeseen/Un-Profiled Test Data (Observables)** - The fault/symptom matrix columns represent coverage of specific tests' measurements. When new or unforeseen/un-profiled test data occurs, the fault/symptom matrix columns will be updated to reflect fault/symptom matrix, and hence, the diagnoses. If a test fails, the potential faults, which could have caused the test to fail, are identified vertically down the column. If new test data becomes

available, or observable, and can be related to a specific fault or set of fault possibilities, then a secondary file can be created/updated which denotes a new column in the matrix. The new column would be identified as a learned or added column. Certainty levels should be added to the possible faults associated with the learned columns. These certainty levels would be different from the fault probability data.

4. **Mismatches in DKB based on Incorrect Design Data Modeling and/or Test Coverage Input** - In some cases, historical data may identify situations where the DKB is wrong based upon design information related to fault propagation flow. This would result in wrong items being indicted based upon test results. Either the fault call-out excludes items, which are possibly faulty or includes items, which are fault-free. This would have occurred from incorrect design input data, which reflected the signal flow connection improperly. The case to be fixed in this situation is where an item, which causes the fault, is not included in the ambiguity group.

VI. DIAGNOSTICS DRIVEN IETM INTEGRATION

The objective of using AIMSS IETM authoring and display package is to develop diagnostics driven IETM. This IETM provides the user with the capability to display maintenance information in text, graphics, and table windows that can be easily manipulated by the technician for preferred viewing. Hypertext and graphic "hot spots" are embedded in descriptions and procedures to provide rapid access to related information contained in the database. For example, while viewing a fault isolation procedure, the technician is provided direct access to related information such as schematics and parts lists via links that are embedded in the text of the procedure and its associated graphics. The AIMSS, a truly interactive authoring system, supports a full complement of dialogs and processes that can be embedded in interactive procedures.

In the Windows environment, the Diagnostician inference engine is a Dynamic Link Library (DLL). It is structured as a library of functions that provide diagnostic services to a client program. Using the AIMSS IETM authoring tool, the Diagnostician's DLL functions were integrated through the Process Editor as "Class V" processes. The "Class V" process acts as a "gateway" to all Diagnostician services. Each "Class V" process interacts with the Diagnostician using Templates. Each template defines a DLL function. Templates are easy to use, provide the function call's exact syntax requirements, and simply require the user to identify specific variables, where applicable.

The Diagnostician contains about forty DLL functions. Of those, the primary ones that will be used for a typical diagnostics driven IETM application are as follows:

Load DKB	Load a System's Diagnostic Knowledge Base
AddData	Input test results to the Diagnostician

GetNextStep	Identify next step to be performed (normally, a test procedure)
GetSuspectCnt	Identify the number replaceable items in the current fault call-out
GetSuspectNames	Identify the name(s) of the replaceable items in the fault call-out
LogData	Log all session history data into the historical data base
EndSession	End the current diagnostic session.

With these basic functions, the diagnostic logic that is authored into the IETM is one "WHILE" loop. The WHILE loop processes as shown in the graphic below. No matter how large the system, or how complex the diagnostic logic, this single WHILE loop (see Figure 2) is all that is needed to incorporate the diagnostic processing with the Diagnostician.

VII. BENEFITS OF REENGINEERING TROUBLESHOOTING PROCEDURES

The objective of this effort was to reduce the maintenance burden of DOD weapon systems by reengineering static TPs of legacy TMs. The objective was achieved with the following benefits:

A. Troubleshooting Procedures replaced by dynamic and more robust diagnostics capability

Diagnostician provides dynamic diagnostic reasoning, instead of static troubleshooting trees. During a maintenance session, the test results can be input to the Diagnostician in any order (no preset sequence) and from any source individually or in combination (including operator observations, test instruments, data bus, data file, built-in test (BIT), automatic test equipment, system panels displays, etc.). Test results can be input as many or as few at a time as the test source(s) can provide (not restricted to one-at-a-time to traverse down a fault tree). The Diagnostician zeroes-in on causes of fault(s) and never

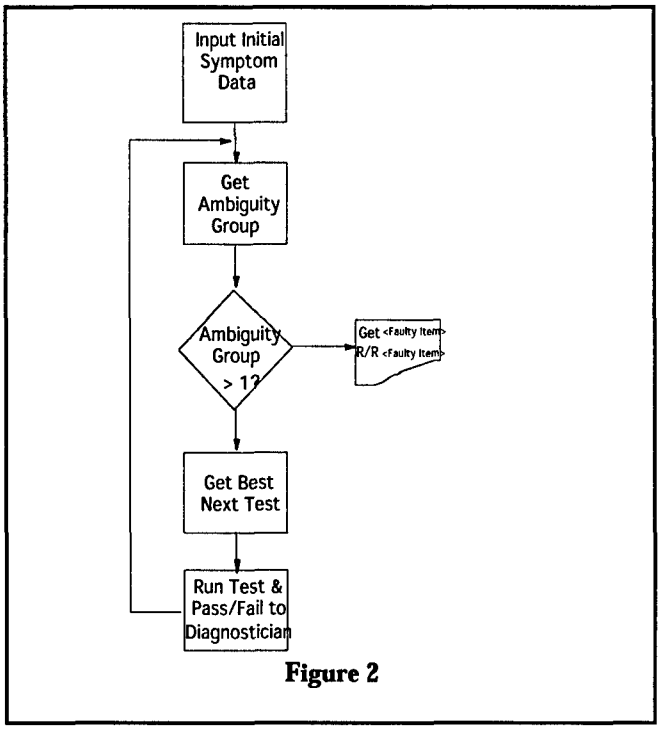

Figure 2

leaves the technician hanging in the middle of a tree! The Diagnostician can identify multiple faults (diagnostic trees follow single-fault assumption). The Diagnostician will only request tests that have diagnostic significance, based upon snapshot of current fault possibilities, and therefore, may decrease overall repair time and increase diagnostics accuracy.

B. Capability of Using Historical Maintenance Data to Continuously Improve Diagnostic

Data Logging of Maintenance History was defined for continuous system diagnostic learning. The Diagnostician creates a log of session profiles. This log is used by the RTS utility to mature diagnostic capability over time. The RTS performs statistical analysis on session history data to identify trends and actual field failure rate data. This is used to automatically or semi-automatically update the DKB and provide improved diagnostics.

C. Built-in Test Data are used for maintenance

Most BIT is designed to support operations, not maintenance, because it focuses on fault detection at the functional level as opposed to diagnostics at the replaceable item level. The Diagnostician can interpret any BIT data and correlate BIT results to a component/item-oriented model of the system. It therefore extends BIT into a maintenance mode to enhance field diagnostics and maintenance.

D. Class V Diagnostics driven IETMs vs. paper TMs

A much more user-friendly IETMs capability was developed, providing the capability to house "expert" diagnostics information in a form that becomes usable and available to novice technicians.

The following are additional benefits for IETM developers:

E. IETM Authoring of Diagnostics is Greatly Simplified

Authors are not required to author the complex "IF THEN...GOTO" logic associated with structured diagnostic trees - all diagnostic logic is inside the DKB. The author simply creates one WHILE loop to manage the dialog with the Diagnostician.

F. Possible Elimination of SGML tag the troubleshooting logic

SGML tagging for diagnostics is very complex and requires extensive content tagging. This can be eliminated by creating the DKB from design data or from an engineering analysis of the troubleshooting trees directly.

VIII. CONCLUSION/SUMMARY

The R&D project to create the capability of reengineering troubleshooting procedures into dynamic model-based diagnostics was successfully completed. The new paradigm for diagnostics, the Diagnostician and the AIMSS IETM authoring and display package worked together very efficiently to accomplish the reengineering task. The troubleshooting procedures from TM for the Army's Fox NBC Reconnaissance vehicle was used as a testbed. A diagnostics driven IETM was successfully completed for Fox. The maintenance school is using it as a training tool. This methodology to reengineer troubleshooting procedures in TMs is generic and can be reused for other legacy systems both from DOD and commercial industry.

REFERENCES

[1] "*Diagnostic Analysis and Repair Tool Set (DARTS): An Enabling Technology for Concurrent Engineering,*" Mary Nolan, Li Pi Su. AUTOTESTCON, San Antonio, September '93

[2] *"Re-Engineering the Test Development Process,"* Paul J. Giordano, Ford Levy. AUTOTESTCON, Anaheim, September '94

[3] *"Intelligent Maintenance Aid Software,"* Gerard Giordano, Greg deMare. AUTOTESTCON, Anaheim, September '94

[4] "*A New Breed of Depot ATE,*" Dave Carey and Paul Giordano. AUTOTESTCON, Atlanta,

August '95

[5] *"An Open Architecture COTS Tester with Intelligent Diagnostics,"* Mary Nolan, Ken Carroll. AUTOTESTCON, Dayton, '96

[6] *"Application of New Information Technology to DOD Legacy Paper Technical Manuals,"* Li Pi Su, Wesley England, Charles D. Bosco. 21st Century Commerce & CALS Expo USA 1997

Biographical Sketch

Dr. Li Pi Su has a B.S. in mathematics, Taiwan Normal University, Taipei, Taiwan, 1959; Ph.D. mathematics, University of British Columbia, Vancouver, Canada, 1966; and B.S. electrical engineering, University of Oklahoma, Norman, Oklahoma, 1983. 1966-67, Post Doctoral Research, University of Toronto. 1967-68, Professor of Mathematics, Tsing-Hua University, Taiwan. 1968-78, Professor of Mathematics, University of Oklahoma. 1980-85, Electrical Engineer, Department of the Air Force (Tinker AFB, OK). 1985-92, Logistics Engineer, Department of the Army (Lexington Army Depot, KY). 1992-Present, Electronic Engineer, Software Engineering Division, U.S. Army, Test, Measurement, and Diagnostic Equipment Activity, Department of the Army (Redstone Arsenal, AL). Since 1992, she has completed management of the research and development projects: the Diagnostic Analysis and Repair Tool Set; Embedded Diagnostics; Reengineering; the Integrated Diagnostics; and Repair Information System Programs. She also has been researching and developing diagnostics/prognostics technology and Interactive Electronic Technical Manual related technology.

Ms. Mary Nolan, Vice President of Systems for Giordano Automation Corp. since 1991, is responsible for applications of the company's diagnostics tools, the Diagnostic Profiler and Diagnostician. She was a Senior Systems Analyst of Giordano Associates, Inc., during 1981-1991. Ms. Nolan has a BS Degree from Trenton State College and also attended the Defense Systems Management College Program Managers Course in Ft. Belvior. Since 1981, she has been working on research and development of automated test and diagnostics, integrated diagnostics, embedded diagnostics, and related areas. She has participated in definition, development, application, and support of an advanced concurrent engineering implementation methodology and associated tools and that facilitates the automation of the "design-to-support" aspect of integrated product development. That resulted in an Expert System Application that provides diagnostic capability at a significantly reduced development and O& S costs. Among many implementations she did on enhancing design for testability and diagnoseability, she participated in the development of MIL-STD-2165. She defined, managed, and participated in implementation of the Diagnostician in several testers at San Antonio Air Logistics Center for Depot Support of F-15 and F-16 repairables. Currently, she is also working on reengineering the troubleshooting procedures from legacy technical manuals into model-based diagnostics driven Interactive Electronic Technical Manuals.

PLEASE CHECK THE APPROPRIATE BLOCK BELOW:

-AQ #_____

☐ _____ copies are being forwarded. Indicate whether Statement A, B, C, D, E, F, or X applies.

☒ DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

☐ DISTRIBUTION STATEMENT B:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES ONLY; (Indicate Reason and Date). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT C:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND THEIR CONTRACTORS; (Indicate Reason and Date). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT D:
DISTRIBUTION AUTHORIZED TO DoD AND U.S. DoD CONTRACTORS ONLY; (Indicate Reason and Date). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT E:
DISTRIBUTION AUTHORIZED TO DoD COMPONENTS ONLY; (Indicate Reason and Date). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT F:
FURTHER DISSEMINATION ONLY AS DIRECTED BY (Indicate Controlling DoD Office and Date) or HIGHER DoD AUTHORITY.

☐ DISTRIBUTION STATEMENT X:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND PRIVATE INDIVIDUALS OR ENTERPRISES ELIGIBLE TO OBTAIN EXPORT-CONTROLLED TECHNICAL DATA IN ACCORDANCE WITH DoD DIRECTIVE 5230.25, WITHHOLDING OF UNCLASSIFIED TECHNICAL DATA FROM PUBLIC DISCLOSURE, 6 Nov 1984 (Indicate date of determination). CONTROLLING DoD OFFICE IS (Indicate Controlling DoD Office).

☐ This document was previously forwarded to DTIC on _____ (date) and the AD number is _____.

☐ In accordance with provisions of DoD instructions, the document requested is not supplied because:

☐ It will be published at a later date. (Enter approximate date, if known).

☐ Other. (Give Reason)

DoD Directive 5230.24, "Distribution Statements on Technical Documents," 18 Mar 87, contains seven distribution statements, as described briefly above. Technical Documents must be assigned distribution statements.

Tom Minnich

per phone call with J. Camp 10/22/98

Authorized Signature/Date

AFRL/FSD
2241 Avionic Circle
Wright-Patterson AFB, OH 45433-7318

John Camp

Print or Type Name

937 255-2164 3562

Telephone Number